DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME 2204 Algorithm Analysis

Assignment-II

by

Mustafa Çağatay ÖNAL

Lecturers

Asst. Prof. Zerrin Işık

Res.Asst. Ali Cüvitoğlu

PhD Student Altuğ Yiğit

IZMIR

17.05.2019

PROGRESS DESCRIPTION

In this assignment, want us to wrote 2 methods. This methods are Dynamic Programming Approach and Greedy Approach. Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. So the problems where choosing locally optimal also leads to global solution are best fit for Greedy. Dynamic programming approach is similar to divide and conquer in breaking down the problem into smaller and yet smaller possible sub-problems. But unlike, divide and conquer, these sub-problems are not solved independently. Rather, results of these smaller sub-problems are remembered and used for similar or overlapping sub-problems.

Fist of all there was a input file, which specified data. This input file format was xlsx, I converted this file to txt file, so my code read data from a txt file. There was four information about each player, so I created a class call Player which based on this information, because read all data and assigned this information to Player array. I created static Player array in main class, so it can be reached inside of any method. So far file is read and assigned in array. After that I am continue with greedy approach because my dynamic programming couldn’t work at all. In greedy method I created temp for Player which specified position number and how much player can get each position. So, find N position player and K player for each position in main array and assigned this player in new array. Our aim is devise an algorithm that maximizes the total rating of the players you sign while spending no more than $x altogether in greedy approach. After created temp array for players, I sorted players in ratio value, which is ratio value was their rating/price value for each player. After that I travelled array from start to end and find best ratio player for each position without exceeding $x value. And that calculate total rating and total price values and print players information which signed.

Running time of greedy approach of my algorithm is O(n^2) which n is player number which specified in n position kth player.

In dynamic programming method, I found pseudocode code of similar problem of our assignment in last day, there was though in my mind all time, all dynamic programming examples or codes or problems there was small data set and small values, our price value can be 150.000 or bigger than that, so I thought it couldn’t be solution of created 2 dimensional array which one dimension was price and other one rating or player number or position vs. It can be literally kill running time of algorithm and memory of computer, because all of 0-1 knapsack problems are created array based on their price, first thought in my mind after that we can be calculate their ratio based on their price value, I mean one more parameter for each player that holds (their price value / total price) \* 100. So that means price value can be 100 and ratio was how much take a place each player in price value. But it’s only stayed on my mind, I couldn’t wrote that code.

REFERENCES

Introduction To Algorithms, Third Edition, Cormen TH, Leiserson CE, Rivest RL, and Stein C. The MIT Press, 2009.

[0/1 Knapsack using Branch and Bound](https://www.youtube.com/watch?v=yV1d-b_NeK8&t=380s)

https://walkccc.github.io/CLRS/Chap15/Problems/15-12/

[Branch and Bound Algorithm](https://www.geeksforgeeks.org/branch-and-bound-algorithm/)

[Dynamic Programming](https://www.geeksforgeeks.org/dynamic-programming/)